EXCEPTION HANDLING

********** EXCEPTION WITH TRY CATCH & ENDTRY BLOCK **********

```
data : res type i.
parameters : num1 type i,
             num2 type i.

data : obj type ref to cx_root.
data : msg type string.
try.
  res = num1 / num2.
  write :/ 'result =', res.
  catch cx_root into obj.
    msg = obj->get_longtext( ).
    write :/ msg.
  endtry.
```

***** EXCEPTION WITH TRY CATCH & ENDTRY BLOCK IN A CLASS *****

```
class lcl_a definition.

  public section.
  data : num1 type i,
           num2 type i,
           res type i.
  data : ref type ref to cx_root,
           msg type string.
  methods : meth importing var1 type I var2 type I.

  endclass.

  class lcl_a implementation.
    method meth.
      num1 = var1.
      num2 = var2.
      try.
      res = num1 / num2.
      write :/ 'result of division = ', res.
      catch cx_root into ref.
        msg = ref->get_longtext( ).
        write :/ msg.
      endtry.
      endmethod.
    endclass.

    start-of-selection.
    data obj type ref to lcl_a.
    create object obj.

    parameters : p_num1 type i,
                 p_num2 type i.
    obj->meth( var1 = p_num1
               var2 = p_num2 ).
```

```
********* EXCEPTION WITH TRY CATCH & ENDTRY BLOCK  ***********

parameters : num type i.
data : RES TYPE P DECIMALS 2,
            o_ref type ref to cx_root,
            text type string.

START-OF-SELECTION.
WRITE :/ 'DIVISION AND SQUARE ROOT CHECKING'.
ULINE.
TRY .
IF ABS( NUM ) > 100.
  RAISE EXCEPTION TYPE CX_DEMO_ABS_TOO_LARGE.
  ENDIF.
TRY.
RES = 1 / NUM.
WRITE :/ 'RESULT OF DIVISION=', RES.
RES = SQRT( NUM ).
WRITE :/ 'RESULT OF SQUARE', RES.
CATCH CX_SY_ZERODIVIDE INTO O_REF.
  TEXT = O_REF->GET_TEXT( ).
  CLEANUP.
    CLEAR RES.
    ENDTRY.
    CATCH CX_SY_ARITHMETIC_ERROR INTO O_REF.
      TEXT = O_REF->GET_TEXT( ).
      CATCH CX_ROOT INTO O_REF.
        TEXT = O_REF->GET_TEXT( ).
        ENDTRY.
        IF NOT TEXT IS INITIAL.
          WRITE :/ TEXT.
          ENDIF.

          WRITE :/ 'FINAL RESULT', RES.
```

******* EXCEPTION WITH NESTED TRY CATCH & ENDTRY BLOCK ******

```
data : o_ref type ref to cx_root,
            text type string.
try.
    try.
        raise exception type cx_demo_constructor
                          exporting my_text = sy-repid.
      catch cx_demo_constructor into o_ref.
        text = o_ref->get_text( ).
        write / text.
        raise exception o_ref.
    endtry.
  catch cx_demo_constructor into o_ref.
    text = o_ref->get_text( ).
    write / text.
endtry.
```

***** EXCEPTION WITH TRY CATCH & ENDTRY BLOCK IN A CLASS *****

```
class lcl_a definition.
public section.
methods : meth1 importing p type string
             raising cx_demo_constructor
                       cx_demo_abs_too_large.

endclass.

class lcl_b definition.
  public section.
  data : obj_a type ref to lcl_a.
  methods : meth2 raising cx_demo_constructor.
  endclass.


    class  lcl_b implementation.
      method meth2.
        create object obj_a.
        try.
          obj_a->meth1( 'tricon' ).
          catch cx_demo_abs_too_large.
          endtry.
        endmethod.
      endclass.

      class lcl_a implementation.
        method meth1.
          raise exception type cx_demo_constructor.
          endmethod.
        endclass.

  data : obj_b type ref to lcl_b.
  start-of-selection.
  create object obj_b.
  try.
    obj_b->meth2( ).
    catch cx_demo_constructor.
      write :/ 'catching cx_demo_constructor'.
    endtry.
```

```
********* EXCEPTION WITH INHERITANCE  IN A CLASS ***********

 class local_exp definition inheriting from
cx_static_check.

  endclass.
  start-of-selection.
  TRY .
raise exception type local_exp.
  CATCH local_exp.
message 'caught' type 'I'.
  ENDTRY.


 ********* EXCEPTION WITH INHERITANCE  IN A CLASS ***********

 class local_exp definition inheriting from
cx_static_check.
  public section.
    data : text type string.
    methods : constructor importing f_text type string.
endclass.

class local_exp implementation.
  method constructor.
    super->constructor( ).
    text = f_text.

  endmethod.
endclass.

data : o_ref type ref to local_exp.
start-of-selection.
TRY .
raise exception type local_exp exporting f_text =
'local exceptions'.
CATCH local_exp into o_ref.
message o_ref->text type 'I'.
ENDTRY.
```

********* EXCEPTION WITH INHERITANCE IN A CLASS **********

```
class cx_local_exception definition inheriting from
cx_sy_arithmetic_error.
  public section.
  methods : constructor importing situation type
string.
  endclass.

  class cx_local_exception implementation.
   method constructor.
     super->constructor( operation = situation ).
     endmethod.
    endclass.

   data : o_ref type ref to cx_local_exception,
          text type string.

    start-of-selection.
    try.
      raise exception type cx_local_exception exporting
situation = 'start-of-selection'.
      catch cx_local_exception into o_ref.
        text = o_ref->get_text( ).
        message text type 'I'.
      endtry.
```

```
**********  EXCEPTION WITH FILES ********************

data : o_ref type ref to cx_sy_file_open_mode,
            text type string.

TRY .
raise exception type cx_sy_file_open_mode
                  exporting textid =
CX_SY_FILE_OPEN_MODE=>read_only
                                        filename = 'SS.DAT'.
CATCH CX_SY_FILE_OPEN_MODE INTO O_REF.
  TEXT = O_REF->GET_TEXT( ).
  MESSAGE TEXT TYPE 'I'.

ENDTRY.

TRY .
raise exception type cx_sy_file_open_mode
                  exporting textid =
CX_SY_FILE_OPEN_MODE=>NOT_OPEN
                                         filename =
'SS.DAT'.
CATCH CX_SY_FILE_OPEN_MODE INTO O_REF.
  TEXT = O_REF->GET_TEXT( ).
  MESSAGE TEXT TYPE 'I'.

ENDTRY.

TRY .
raise exception type cx_sy_file_open_mode
                  exporting textid =
CX_SY_FILE_OPEN_MODE=>INCOMPATIBLE_MODE
                                           filename =
'SS.DAT'.
CATCH CX_SY_FILE_OPEN_MODE INTO O_REF.
  TEXT = O_REF->GET_TEXT( ).
  MESSAGE TEXT TYPE 'I'.

ENDTRY.
```